

22<sup>o</sup>-1552

Syed Arham Ahmed

BS-CY-B

## COAL Assignment #1

**DUE DATE: 24<sup>th</sup> Feburary, 2024 11:59 PM**

Print this assignment and complete it on the printed sheet only. No extra sheet is allowed. Scan/Camscan the solution and upload it on Google Classroom.

You can verify the answers using MASM debugging mode.

You may need to use your roll number in some of the problems. The respective digits of your roll number to be used will be enclosed in  $\langle \rangle$ . So, for example, if your roll number is i12-3456 then the first digit is '1' and the sixth digit is '6'. So  $\langle 1^{\text{st}}, 3^{\text{rd}}, 4^{\text{th}}, 2^{\text{nd}} + 5^{\text{th}} \rangle$  means 1347h.

### Question # 01

Perform each of the following operations on word size 2's complement numbers and ~~upl~~the answer and value of flags after performing the arithmetic. Perform all the steps in the "calculation" box, only filling the answer will not get any credit.

1.  $\text{FFFE} + \langle 1^{\text{st}} + 6, 2^{\text{nd}} + 6, 5^{\text{th}} + 6, 6^{\text{th}} \rangle$        $\text{FFFE} + 8882 = 8880$

Answer	
Flags	Sign
	Zero
	Carry
	Overflow
	Aux Carry
	Parity

Calculation	1111	1111	1111	1110
	+ 1000	1000	1011	0010
①	1000 1000 1011 0000			

2.  $8080 + \langle 2^{\text{nd}}, 1^{\text{st}}, 6^{\text{th}}, 4^{\text{th}} \rangle$

$8080 + 2225 = \text{A2AS}$

Answer	
Flags	Sign
	Zero
	Carry
	Overflow
	Aux Carry
	Parity

Calculation	1000	0000	1000	0000
	0010	1010	0010	0101
	1010 0010 1010 0101			

3.  $\langle 3^{\text{rd}}, 3^{\text{rd}}, 6^{\text{th}}, 5^{\text{th}} \rangle - 71\text{FF}$

1125 - 71FF = FFFF9F26

Answer	
Sign	1
Zero	0
Carry	0
Overflow	0
Aux Carry	0
Parity	0 odd

Calculation	0001 0001 0010 0101
	1000 1110 0000 0001
	1001 1111 0010 0110

4.  $4785 + \langle 3^{\text{rd}} + 6, 2^{\text{nd}} + 6, 3^{\text{rd}} + 6, 1^{\text{st}} \rangle$

4785 + 7872 = BFF7

Answer	
Sign	1
Zero	0
Carry	0
Overflow	1
Aux Carry	0
Parity	1 even

Calculation	0100 0111 1000 0101
+	0111 1000 0111 0010
	1011 1111 1111 0111

5.  $64\text{EA} + \langle 1^{\text{st}} + 6, 3^{\text{rd}} + 6, 2^{\text{nd}} + 6, 4^{\text{th}} \rangle$  64EA + 8785 = EC65

Answer	
Sign	1
Zero	0
Carry	0
Overflow	0
Aux Carry	0
Parity	0 odd

Calculation	0110 0100 1110 1010
+	1000 0111 1000 0101
	1110 1100 0110 1111

6.  $\text{EEB2} - \langle 1^{\text{st}} + 5, 2^{\text{nd}}, 4^{\text{th}}, 6^{\text{th}} \rangle$

EEB2 - 7252 = 7C60

Answer	
Sign	0
Zero	0
Carry	1
Overflow	1
Aux Carry	0
Parity	0 odd

Calculation	1000 1101 1010 1110
+	1110 1110 1011 0010
①	0111 1100 0110 0000

8100 + 2252 = A352

7.  $8100 + \langle 2^{\text{nd}}, 1^{\text{st}}, 4^{\text{th}}, 2^{\text{nd}} \rangle$

Answer	
Sign	0
Zero	0
Carry	0
Overflow	1
Aux Carry	0
Parity	1 even

Calculation	1110 1110 1110 1110
+	0100 0101 0010 0010
①	0011 0100 0001 0000

$$\begin{array}{r} \text{EEEE} + 4527 \\ = 13410 \end{array}$$

8. EEEE + <1<sup>st</sup>, 2<sup>nd</sup>, 4<sup>th</sup>, 6<sup>th</sup>, 2<sup>nd</sup>>

Answer	
Sign	0
Zero	0
Carry	1
Overflow	1
Aux Carry	1
Parity	1 even

Calculation	1110 1110 1110 1110
	+ 0100 0101 0010 0010
①	0011 0100 0001 0000

### Question # 02

Consider the following Data declaration? Fill given Memory for following declaration?

NOTE: First Data memory address starts at 0x0000

```
valb_1 BYTE 'A'
valb_2 BYTE 0
valb_3 DB 255
valb_4 BYTE -128
valb_5 BYTE +127
valb_6 SBYTE -128
valb_7 SBYTE +127
```

0000	41
0001	0
0010	FF
0011	80
0100	7F
0101	80
0110	7F
0111	
1000	
1001	
1010	
1011	
1100	

```
1111 1111 1111 1111 1111 1111 1000 0000
0000 0000 0000 0000 0000 0000 0111 1111
+
-----
0000 0000 0000 0000 0000 0000 1000 0000
```

Question # 03

Consider the following Data declaration? Fill given Memory?

```
List1 BYTE 1,2,3,4  
       BYTE 5,6,7,8  
       BYTE 9,10,11,12
```

```
List2 DB 10, 04lh, 'A', 00111111b
```

```
string BYTE 'ABC',0  
string2 BYTE "abc",0
```

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	01	02	03	04	05	06	07	08	09	0A	0B	0C	0A	41	41	3F
0010	41	42	43	00	61	62	63	00								
0020																



#### Question # 04

Consider the following Data declaration? Fill given Memory?

```
list1 BYTE 1,2
list2 BYTE 3 DUP(2,0)
list3 DB 2 DUP(?,1)
list4 BYTE 2 DUP(0ABH)
list5 BYTE 3 DUP('AB')
```

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	01	02	02	00	02	00	02	00	?	1	?	1	AB	AB	41	42
0010	41	42	41	42												
0020																

#### Question # 05

Consider the following Data declaration? Fill given Memory?

.data

```
word1 DW 1
word2 DW -1
word3 SWORD -1
word3 SWORD +1
word2 WORD 01F2BH, 01101011001011b, 45
list1 DW 10,2
list2 WORD 3 DUP(0)
list3 DW 2 DUP(?)
list4 WORD 2 DUP(0AB12H)
list5 DW 3 DUP('AB')
```

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	01	00	FF	FF	FF	FF	01	00	2B	1F	CB	1A	2D	00	0A	00
0010	02	00	00	00	00	00	00	?	00	?	00	12	AB	12	AB	42
0020	41	42	41	42	41											

### Question # 06

Consider the following Data declaration? Fill given Memory?

```
.data
quad1 DQ 0123456789ABCDEFH
quad2 QWORD -1
quad3 QWORD 'AB'
quad4 QWORD 2 DUP(012AB34CDh)
```

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	EF	CD	AB	89	67	45	23	01	FF	FF	FF	FF	FF	FF	FF	FF
0010	42	41	00	00	00	00	00	00	CD	34	AB	12	00	00	00	00
0020	CD	34	AB	12	00	00	00	00								

### Question # 07

Consider the following Data declaration? Fill given Memory?

```
.data
word2 DW -1
list1 BYTE 1,2
quad1 DQ 0123456789ABCDEFH
List2 DB 10, 041h, 'A', 00111111b
string BYTE 'ABC',0
list4 WORD 2 DUP(0AB12H)
quad3 QWORD 'AB'
```

1 data = 8 byte

dq = quad word

word = 2 byte

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	FF	FF	01	02	EF	CD	AB	89	67	45	23	01	0A	41	41	3F
0010	41	42	43	00	12	AB	12	AB	42	41	00	00	00	00	00	00
0020																

### Question # 08

Declare Data in following order? Submit Data declaration and memory snapshot with assignment?

- Your Name in Capital Letters: As String ✓
- Your Name in Small Letters: As String ✓
- Your ID in HEX: using Declaration type as WORD ✓
- Your ID in HEX: using Declaration type as DWORD ✓
- Your ID in BIN: using Declaration type as WORD ✓
- Your ID in BIN: using Declaration type as DWORD ✓
- Your ID in DEC: using Declaration type as WORD ✓
- Your ID in DEC: using Declaration type as DWORD ✓
- Your Age in BIN: using Declaration type as BYTE ✓
- Your Age in DEC: using Declaration type as BYTE ✓
- Your Age in BIN: using Declaration type as BYTE ✓
- Your CNIC in HEX: using Declaration as QWORD (Ignore – in your ID CARD Number) ✓

• data

namecaps byte 'ARHAM', 0

name small byte 'arham', 0

idword word 022h, 01552h

id dword dword 0221552h

idbinword word 0010 0010 b, 0001 0101 0101 0010 b

idbindword dword 0010 0010 0001 0101 0101 0010 b

idworddec word 34, 5458

idworddec dword 2233682

agebinbyte byte 0001 0011 b

age decbyte byte 19

agebinbyteagain byte 0001 0011 b

cnichex qword 03D4 3FE4 6E7F H



22 1552

Question # 09

25

Fill the value of respective register after every operation. At start the value of  $AX=0$ , then update the  $AX$  register accordingly afterwards. All the values should be written in Hexadecimal 2s complement Format.

1. MOV AX, <3<sup>rd</sup>, 5<sup>th</sup>>d

AH	AL
00	0F

2. MOV AH, <1<sup>st</sup>, 4<sup>th</sup>>

25

AH	AL
19	0F

3. MOV AH, 1001b

AH	AL
09	0F

4. MOV AL, <2<sup>nd</sup> + 4, 5<sup>th</sup>>d

2 + 4 = 6, 5

AH	AL
09	41

5. MOV AH, 1100b

AH	AL
0C	41

6. MOV AX, <2<sup>nd</sup>, 4<sup>th</sup>, 6<sup>th</sup>>H

2, 5, 2

AH	AL
02	52

7. MOV AH, - <1<sup>st</sup>, 6<sup>th</sup>>

2, 2

AH	AL
EA	52

8. MOV AX, - <3<sup>rd</sup>, 6<sup>th</sup>>

1, 2

AH	AL
00	F4

9. MOV AH, 10000000b

AH	AL
80	F4

10. MOV AH, 1010H

AH	AL
Memory	mismatch error

$$\begin{array}{r}
 0001 \quad 0110 \\
 1110 \quad 1001 \\
 + \quad \quad \quad 1 \\
 \hline
 1110 \quad 1010
 \end{array}$$

$$\begin{array}{r}
 0000 \quad 1100 \\
 1111 \quad 0011 \\
 + \quad \quad \quad 1 \\
 \hline
 11110100
 \end{array}$$



22 1552

**Question # 10**

Find the values of respective registers in HEXADECIMAL

<sup>2 2 2 2</sup> <sup>2 5 2 2</sup> Var01 dword <1 <sup>st</sup> , 1 <sup>st</sup> , 2 <sup>nd</sup> , 6 <sup>th</sup> >9890H, 8723<6 <sup>th</sup> , 4 <sup>th</sup> , 1 <sup>st</sup> , 2 <sup>nd</sup> >H, 11223344H	
1. mov AX, word PTR [Var01+5]	AX = 2325
2. mov AX, word PTR [Var01+9]	AX = 2233
3. mov AX, word PTR [Var01+10]	AX = 1122
4. mov AL, byte PTR [Var01+7]	AL = 87
5. mov AL, byte PTR [Var01+8]	AL = 44
6. mov AL, byte PTR [Var01+4]	AL = 22
7. mov AL, byte PTR Var01	AL = 90
<sup>1 5</sup> <sup>2 5</sup> <sup>2 5</sup> <sup>2 5</sup> Var02 byte <3 <sup>rd</sup> , 5 <sup>th</sup> >H, <2 <sup>nd</sup> , 4 <sup>th</sup> >H, <6 <sup>th</sup> , 4 <sup>th</sup> >H, <1 <sup>st</sup> , 5 <sup>th</sup> >H, 90H, 1AH, 0DH, 0FFH, 0EH, 0CCH	
1. MOV AX, Word PTR [Var02 + 2]	AX = 2525
2. MOV EAX, Dword PTR [Var02 + 8]	EAX = 00EHFF00
3. MOV EAX, Dword PTR [Var02 + 5]	EAX = FF000D1A
4. MOV AX, Word PTR [Var02 + 9]	AX = EH00
5. MOV AX, Byte PTR Var01	AX = Memory mismatch
<sup>2 2 1 5</sup> <sup>5 1 2 2</sup> <sup>5 2 1 5</sup> <sup>1 1 2</sup> Var03 word <1 <sup>st</sup> , 2 <sup>nd</sup> , 3 <sup>rd</sup> , 4 <sup>th</sup> >H, <4 <sup>th</sup> , 3 <sup>rd</sup> , 1 <sup>st</sup> , 2 <sup>nd</sup> >H, <5 <sup>th</sup> , 6 <sup>th</sup> , 3 <sup>rd</sup> , 4 <sup>th</sup> >H, <3 <sup>rd</sup> , 3 <sup>rd</sup> , 1 <sup>st</sup> , 4 <sup>th</sup> >H, 90ABH, 0BA09H	
6. MOV AL, Word ptr [Var03 + 3]	AL = Memory mismatch
7. MOV EAX, Dword ptr [Var03 + 5]	EAX = AB112552
8. MOV EAX, Dword ptr [Var03 + 8]	EAX = BA0990AB
9. MOV AH, Word ptr [Var03 + 10]	AH = Memory mismatch
10. MOV AH, Byte ptr Var03	AH = 15

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
90	08	22	22	22	25	23	87	44	33	22	11				

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
15	25	25	25	90	1A	0D	00	FF	00	EH	00	CC			

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
15	22	22	51	15	52	25	11	AB	90	09	BA				

Assignment 1  
Computer Organization and Assembly Language (spring 2024 )

Question # 11

Update registers after executing following piece of code

```
mov ah, 0111b
mov al, 01110111b
mov bx, 0AB00h
mov bl, 0CDh
mov cx, 512 ✓
mov cl, 254 ✓
mov dx, 760 ✓
mov dh, 230 ✓
```

AX	07	77
BX	AB	00 CD
CX	02	00 FE
DX	7013	3E

← CD after

Question # 12

Update registers after executing following code.

```
.data
    bval BYTE 5
    wval WORD 012h
    result word ?
.code
    mov al, bval
    mov bx, wval
    add bx, ax 0017
    mov result, bx result, 0017
    mov dl, bl
    mov cx, result
```

	ah	al
AX	00	05
BX	0000	2E 17
CX	00	17
DX	00	17

00 05

**Assignment 1**  
**Computer Organization and Assembly Language (spring 2024 )**

**Question # 13**

Rewrite following code after removing errors. Also mention type of error

```

5  .data
6      bval1 BYTE 0
7      bval2 BYTE 0
8      wval WORD 0
9      result word ?
10 .code
11     mov ax,bval1
12     mov bl,wval
13     add bval1,bval2
14     mov 3,bval2
15     mov 4,ax
16     mov cl,bval
17     mov result,cl

```

```

• data
bval1 byte 0
bval2 byte 0
wval word 0
result word ?

• code
mov al, bval1 [ 8 to 16 bit not possible ]
mov bx, wval [ 16 to 8 bit not // ]
add al, bval2 [ memory to memory not ]
mov bval2, 3 [ destination/register first ]
mov ax, 4 [ // // ]
mov cl, bval2 [ bval not exist ]
mov result, ax [ memory mismatch ]

```

**Assignment 1**  
**Computer Organization and Assembly Language (spring 2024)**

**Question # 14**

Update registers and memory after executing following code.

```

6      bval1 BYTE 5
7      bval2 BYTE 7
8      wval1 WORD 0ABCDH
9      wval2 word 01234h
10     .code
11     mov al,bval1
12     mov bl,bval2
13     xchg al,bl
14     mov bval1,bl
15     mov bval2,al
16     mov cx,wval1
17     xchg cx,wval2
18     mov wval2,cx

```

Memory Before Execution

	00	01	02	03	04	05	06
0100	05	07	CD	AB	34	12	

Memory After Execution

	00	01	02	03	04	05	06
0100	05	07	CD	AB	34	12	

	eax	ebx	ecx
AX			05 07
BX			01 05
CX	00 34	00 12	00 00
DX	00	00	

al = 07

bl = 05

wval2 = ~~24 00 12 34~~

~~34 12 00 00~~

00 34 12 34h



Assignment 1  
Computer Organization and Assembly Language (spring 2024 )

---

**Question # 15**

Update Registers after following code

```
5 .data.  
6     bary1 BYTE 5,4,3,2,1  
7     bary2 BYTE 9,0AH,0BH,0CH  
8     wary1 WORD 1,2,3  
9     wary2 word 0AaAaH,0BbBbH,0CcCcH  
10 .code  
11     mov ax,@data  
12     mov ds,ax  
13     mov ax,0  
14     mov al,bary1  
15     add al,bary2  
16     mov ah,[bary1+1]  
17     add ah,[bary2+1]  
18     mov bh,[bary1+2]  
19     add bh,[bary2+2]  
20     mov bl,[bary1+5]  
21     add bl,[bary2+5]  
22     mov cx,wary1  
23     mov dx,[wary1+2]  
24     mov si,[wary1+4]  
25     add cx,wary2  
26     add dx,[wary2+2]  
27     add si,[wary2+4]
```

AX	0E	0E
BX	0E	09
CX	AA	AB
DX	BB	BD
Si	CC FF	

**Assignment 1**  
**Computer Organization and Assembly Language (spring 2024)**

**Question # 16**

Update SI register after every instruction? Consider bval1 is at address 0x0100

```

7      bval1  DB  1
8      wval2  DW  2
9      dval3  DD  3
10     val4   DB  0abH
11     val5   DB  0abH
12     ary    dw  5 dup(0)
13     pary   dw  ary

```

0100	01	02	00	03	00	00	00
0108	0b	0b	0	0	0	0	0
0109							

```

mov si, OFFSET bval1
mov si, OFFSET [bval1+1]
mov si, OFFSET wval2
mov si, OFFSET dval3
mov si, pary
mov si, OFFSET ary

```

Si	0100
Si	0101
Si	0101
Si	0103
Si	0109
Si	0109

**Question # 17**

Consider following data declaration and fill memory and update SI after execution of every instruction? Address starts at 0x0100

```

bval1 DB 1,2,3
wval2 WORD 0ABCDH
bval3 DB 'A'
wval4 DW 01234H
bval5 byte 'B'

```

			3		5				
0100	01	02	03	CD	AB	41	34	12	
0108	42								

```

mov si, OFFSET bval1
mov si, OFFSET wval2
mov si, OFFSET bval3
mov si, OFFSET wval4
mov si, OFFSET bval5

```

Si	0100
Si	0103
Si	0105
Si	0106
Si	0108

**Assignment 1**  
**Computer Organization and Assembly Language (spring 2024 )**

**Question # 18**

Consider following data declaration and fill memory and update SI after execution of every instruction.  
 Observe difference in SI values from question 17? Observe working of Align Directive

```
bval1 DB 1,2,3
Align 2
wval2 WORD 0ABCDH
bval3 DB 'A'
Align WORD
wval4 DW 01234H
bval5 byte 'B'
```

0100	01	02	03	00	CD	AB	41	00
0108	34	12	42					
0110								

```
mov si, OFFSET bval1
mov si, OFFSET wval2
mov si, OFFSET bval3
mov si, OFFSET wval4
mov si, OFFSET bval5
```

Si	0100
Si	0104
Si	0106
Si	0108
Si	010A

**Question # 19**

Update value of registers after each line of code

```
bval1 DB 034h,012h
wval2 DW 0ABCDH
dval3 DD 0ABCDEF12H
```

```
mov ax,WORD PTR bval1
mov al,BYTE PTR wval2
mov bx,WORD PTR dval3
mov cx,WORD PTR [dval3+2]
```

AX	12	34
AL	—	CD
BX	EF	12
CX	AB	CD

Assignment 1  
Computer Organization and Assembly Language (spring 2024 )

---

**Question # 20**

Update value of registers after every instruction

```
bval1 DB 012h  
wval2 DW 0ABCDH  
dval3 DD 0ABCDEF12H  
qval4 QWORD 01243567812435678H
```

```
mov al, TYPE bval1  
mov bl, TYPE wval2  
mov cl, TYPE dval3  
mov dl, TYPE qval4
```

Al	01
BL	02
CL	04
DL	08



**Assignment 1**  
**Computer Organization and Assembly Language (spring 2024)**

---

**Question # 21**

Update value of registers after every instruction

```
bval1 DB 012h
wval2 DW 0ABCDH
dval3 DD 0ABCDEF12H
qval4 QWORD 01243567812435678H
```

```
mov al, TYPE bval1
mov bl, TYPE wval2
mov cl, TYPE dval3
mov dl, TYPE qval4
```

AI  
BL  
CL  
DL

01
02
04
08

**Question # 22**

Update value of registers after every instruction.

```
bAry BYTE 010H,020h,030H
wAry WORD 5 DUP(?),0,0
string1 BYTE 0FFH,0FFH
dAry DWORD 2 DUP(3 DUP(0)),0
string2 BYTE 'ABCDEFGHIJKLMNOPQRSTUVWXYZ',0
bAry2 BYTE 1,2,3,
         4,5,6
Byte     4,5,6
```

```
mov bl, LENGTHOF bAry
mov cl, LENGTHOF wAry
mov dl, LENGTHOF dAry
mov ah, LENGTHOF string2
mov al, LENGTHOF bAry2
```

bl	03
cl	07
dl	07
ah	1B
al	06

**Assignment 1**  
**Computer Organization and Assembly Language (spring 2024 )**

---

**Question # 23**

Find the values of SizeOf, LengthOf and Type operators for the respective variables.

.data	SizeOf	LengthOf	Type
V1 byte 11,22,33,44,55,66	06	06	01
V2 word 15 Dup (0), 5,7,10	36	18	02
V3 dword 4 DUP(10 Dup(4))	160	40	04
V4 word 1,2,4,5,6,2, 7,8,9,9 Word 7,8,5,6,8,9,8 ?	20	10	02
V5 qword "Hello World!!!", 0	24	3	08

$$15 \times 2 = 30 + 2 + 2 + 2 = 36$$

40 x 4

160